(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 07020778 A

(43) Date of publication of application: 24.01.95

(51) Int. Cl G09C 1/00
G06F 7/52
G06F 7/72

(21) Application number: 05164870

(22) Date of filing: 02.07.93

(71) Applicant: FUJITSU LTD

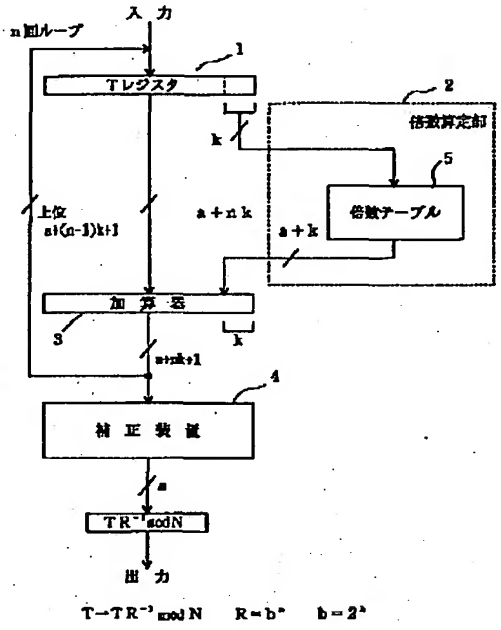
(72) Inventor: TAKENAKA MASAHIKO
TORII NAOYA
HASEBE TAKAYUKI
AKIYAMA RYOTA

(54) REMAINDER CALCULATING DEVICE, TABLE GENERATING DEVICE, AND MULTIPLICATION REMAINDER CALCULATING DEVICE

(57) Abstract:

PURPOSE: To enable a high-speed processing by simplifying an arithmetic processing for remainder calculation.

CONSTITUTION: A multiple table 5 containing multiplies of a modulus N of a remainder corresponding to prescribed low-order bits of an input register 1 is provided and retrieved on the basis of prescribed low-order bits of a number to be subjected to the remainder calculation T in the input register 1 to obtain a multiple of the modulus N of the remainder corresponding to the prescribed low-order bits of T. An adder 3 adds the multiple of the modulus N of the remainder, obtained by the indexing in the multiple table 5, to the contents of the input register 1. The contents of the input register 1 are updated with the prescribed high-order bits of the addition result of the adder 3 each time the addition by the adder 3 is performed (n) times as prescribed. A correcting device 4 performs a correcting processing for (t) obtained as the addition result of the adder 3.



改案

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平7－20778

(43)公開日　平成７年(1995)１月24日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０９Ｃ	1/00		8837－5L		
Ｇ０６Ｆ	7/52		A		
	7/72				

審査請求　未請求　請求項の数10　ＯＬ　（全 16 頁）

(21)出願番号　特願平5－164870

(22)出願日　平成５年(1993)７月２日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中1015番地

(72)発明者　武仲　正彦
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　鳥居　直哉
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　長谷部　高行
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74)代理人　弁理士　遠山　勉　（外１名）

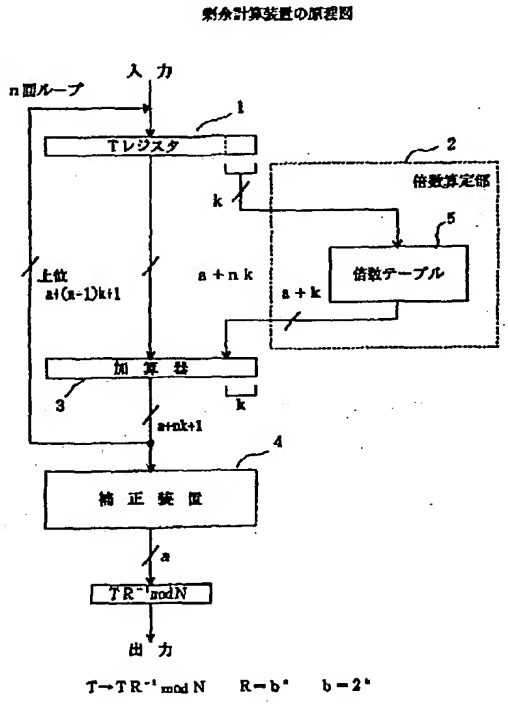
最終頁に続く

(54)【発明の名称】　剰余計算装置、テーブル作成装置および乗算剰余計算装置

(57)【要約】

【目的】剰余計算の演算処理を簡略化して、高速処理を実現する。

【構成】入力レジスタ１の下位所定ビットに対応して剰余の法Ｎの倍数が格納された倍数テーブル５を有し、入力レジスタ１の被剰余数Ｔの下位所定ビットに基づいて倍数テーブル５を検索することにより、前記Ｔの下位所定ビットに対応する剰余の法Ｎの倍数を求める。加算器３は、倍数テーブル５の索引により求められる剰余の法Ｎの倍数と入力レジスタ１の内容とを加算する。加算器３の所定回数ｎの加算について、加算器３の加算結果の上位所定ビットで入力レジスタ１の内容を更新する。補正装置４は、加算器３の加算結果として得られるｔに対して補正処理を施す。



【特許請求の範囲】

【請求項１】　入力される被剰余数を保持する入力レジスタ（１）と、

前記入力レジスタ（１）に格納される被剰余数の下位所定ビットに基づいて、予め演算データを格納したテーブルを用いるテーブル検索により、前記被剰余数の下位所定ビットに対応する剰余の法の倍数を求めるための倍数算定手段（２）と、

前記倍数算定手段（２）で求められる剰余の法の倍数と前記入力レジスタ（１）の内容とを加算するとともに、所定回数の加算についてその加算結果の上位所定ビットで前記入力レジスタ（１）の内容を更新するための加算手段（３）と、

前記加算手段（３）の加算結果に対して剰余結果の補正処理を施すための補正手段（４）とを具備することを特徴とする剰余計算装置。

【請求項２】　倍数算定手段（２）は、被剰余数の下位所定ビットに対応させて剰余の法の倍数を格納した倍数テーブル（５）を有し、この倍数テーブル（５）を前記被剰余数の下位所定ビットで検索して、それに対応する前記剰余の法の倍数を直接求める手段であることを特徴とする請求項１に記載の剰余計算装置。

【請求項３】　倍数算定手段（２）は、被剰余数の下位所定ビットに対応させて倍数情報を格納した倍数情報テーブル（４１）を前記被剰余数の下位所定ビットで検索して、それに対応する倍数情報を求める倍数情報算定手段（４２）と、剰余の法を格納する剰余の法レジスタ（１２）と、前記倍数情報算定手段（４２）で得られる倍数情報と前記剰余の法レジスタ（１２）から得られる剰余の法とを乗算する乗算手段（１４）とを有し、前記倍数算定手段（１０）により前記被剰余数の下位所定ビットに対応する前記剰余の法の倍数を求める手段であることを特徴とする請求項１に記載の剰余計算装置。

【請求項４】　請求項２の剰余計算装置に用いる倍数テーブルを記憶装置上に作成するテーブル作成装置において、剰余の法を繰り返し加算するための繰り返し加算手段（６２）と、前記繰り返し加算手段（６２）で求められる各加算結果の値の下位ｂビットの２の補数を求めるための補数算定手段（６４）と、前記補数算定手段（６４）で得られる各補数値に対応するテーブルアドレスに前記各加算結果をそれぞれ格納するためのテーブル書き込み手段とを具備することを特徴とするテーブル作成装置。

【請求項５】　被乗数の下位から所定ビット毎の値と乗数を乗算し、その乗算結果に剰余を加算する分割乗算装置（７５）と、

この分割乗算装置（７５）から出力される被剰余数を保持する被剰余数レジスタ（１１）と、

前記被剰余数レジスタ（１１）に格納される被剰余数の下位所定ビットに基づいて、予め演算データを格納したテーブルを用いるテーブル検索により、前記被剰余数の下位所定ビットに対応する剰余の法の倍数を求めるための倍数算定手段と、

前記倍数算定手段で求められる剰余の法の倍数と前記入力レジスタの内容とを加算するとともに、所定回数の加算についてその加算結果を前記剰余として前記分割乗算装置の前記乗算結果への加算に供するための加算手段（１６）と、

前記加算手段（１６）の加算結果に対して剰余結果の補正処理を施すための補正手段（２２）とを有することを特徴とする乗算剰余計算装置。

【請求項６】　倍数算定手段は、被剰余数の下位所定ビットに対応させて剰余の法の倍数を格納した倍数テーブル（５１）を有し、この倍数テーブル（５１）を前記被剰余数の下位所定ビットで検索して、それに対応する前記剰余の法の倍数を直接求める手段であることを特徴とする請求項５に記載の乗算剰余計算装置。

【請求項７】　倍数算定手段は、被剰余数の下位所定ビットに対応させて倍数情報を格納した倍数情報テーブル（４１）を前記被剰余数の下位所定ビットで検索して、それに対応する倍数情報を求めるための倍数情報算定手段と、剰余の法を格納する剰余の法レジスタ（１２）と、前記倍数情報算定手段で得られる倍数情報と前記剰余の法レジスタ（１２）から得られる剰余の法とを乗算するための乗算手段（１４）とを有し、この倍数算定手段により前記被剰余数の下位所定ビットに対応する前記剰余の法の倍数を求める手段であることを特徴とする請求項５に記載の乗算剰余計算装置。

【請求項８】　整数ＴおよびＮについて、被剰余数Ｔに対する剰余の法Ｎにおける剰余ＴmodＮ＝ＲＥＤＣ（Ｔ＊（R^2modＮ）を求めるためＴR^{-1}modＮ＝ＲＥＤＣ（Ｔ）を算出する剰余計算装置において、

入力される被剰余数Ｔを保持する入力レジスタ（１１）と、

前記入力レジスタ（１１）に格納される被剰余数Ｔの下位所定ビットに基づいて、予め演算データを格納したテーブルを用いるテーブル検索により、前記被剰余数の下位所定ビットに対応する剰余の法Ｎの倍数ｍ′Ｎ（但し、ｍ′＝（Ｔmodｂ）Ｎ′modｂ、ここでＮ′およびＲは整数であり、Ｒ＝b^n、ｂ＝2^kとする）を求めるための倍数算定手段と、

前記倍数算定手段で求められる剰余の法Ｎの倍数ｍ′Ｎと前記入力レジスタの内容とを加算するとともに、所定回数の加算についてその加算結果の上位所定ビットで前記入力レジスタの内容を更新することによりｔ＝（Ｔ＋ｍＮ）／Ｒを出力する加算器（１６）と、

前記加算器（１６）の加算結果ｔに対してｔ＜Ｎの場合はそのままｔ＝ＲＥＤＣ（Ｔ）、ｔ≧Ｎの場合はｔ－Ｎ＝ＲＥＤＣ（Ｔ）として補正処理を施しＴR^{-1}modＮを得る補正装置（２２）とを有することを特徴とする剰余

計算装置。

【請求項９】　倍数算定手段（５２）は、被剰余数Ｔの下位所定ビットに対応させて剰余の法Ｎの倍数ｍ′Ｎを格納した倍数テーブル（５１）を有し、この倍数テーブル（５１）を前記被剰余数Ｔの下位所定ビットで検索して、それに対応する前記剰余の法の倍数ｍ′Ｎを直接求める手段であることを特徴とする請求項８に記載の剰余計算装置。

【請求項１０】　倍数算定手段は、被剰余数Ｔの下位所定ビットに対応させて倍数情報ｍ′を格納した倍数情報テーブル（４１）を前記被剰余数Ｔの下位所定ビットで検索して、それに対応する倍数情報ｍ′を求める倍数情報算定手段（４２）と、剰余の法Ｎを格納する剰余の法レジスタ（１２）と、前記倍数情報算定手段（４２）で得られる倍数情報ｍ′と前記剰余の法レジスタ（１２）から得られる剰余の法Ｎとを乗算する乗算器（１４）とを有し、この倍数算定手段により前記被剰余数の下位所定ビットに対応する前記剰余の法Ｎの倍数ｍ′Ｎを求める手段であることを特徴とする請求項８に記載の剰余計算装置。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、例えば公開鍵暗号系におけるＲＳＡ暗号処理における剰余演算等に好適な剰余計算装置に係り、特に剰余計算の一手法であるモンゴメリのアルゴリズム（Modulo Multiplication Without Trial Division, Peter L. Montgomery, Mathematics of Computation, Volume 44, Number 170, April 1985 pp. 519～528参照）を用いて、高速に剰余計算を行う剰余計算装置および乗算剰余計算装置ならびに剰余計算に使用するための倍数テーブルを作成するテーブル作成装置に関する。

【０００２】

【従来の技術】近年のコンピュータネットワークの発達により、データベースの検索や電子メール、電子ニュースなどの電子化された情報をネットワークを経由して送受する機会が急速に増加してきている。さらに、これらを利用して、オンラインショッピングなどのサービスも提供されつつある。しかし、それにともなって、ネットワーク上の電子化されたデータを盗聴する、改ざんする、他人になりすましてサービスを受けるなどの問題も指摘されている。特に無線を利用したネットワークにおいては、傍受が容易なためにこれらを防止する対策が望まれている。

【０００３】これらの問題に対して暗号技術(encryption technology)を応用した暗号化電子メールや利用者認証システムが提案され、種々のネットワークにも導入されつつある。この意味でコンピュータネットワークにおいては暗号化は必須の技術であるといえる。このような暗号技術の中の１つにディジタル署名すなわち認証に適した公開鍵暗号方式(public key cryptosystem)があるが、暗号化／復号に大量の処理が必要なため高速化が望まれており、様々な高速化アルゴリズムも発表されている。

【０００４】暗号化方式は、大別すると秘密鍵暗号系と公開鍵暗号系の２つに分類することができる。秘密鍵暗号系は、送信者と受信者が同じ暗号鍵を持つことにより暗号通信を行う方式である。すなわち、秘密暗号系では、あるメッセージを秘密の暗号鍵に基づいて暗号化して相手に送り、受け手はこの暗号鍵を用いて暗号を復号しもとのメッセージに戻して情報を入手する。

【０００５】公開鍵暗号系は、送信者は公開されている受信者の公開鍵でメッセージを暗号化して送信し、受信者は自分の秘密鍵でその暗号化メッセージを復号することで通信を行う方式である。すなわち、公開鍵暗号系では、公開鍵は暗号化のための鍵、秘密鍵は公開鍵により暗号化された暗号を復号するための鍵であり、公開鍵で暗号化した暗号は秘密鍵でのみ復号することができる。

【０００６】秘密鍵暗号系では、個人が秘密に保管しなければならない鍵の数が通信相手の数だけ必要であり、必要な総鍵数はｎ人のネットワークの場合ｎ（ｎ－１）／２個である。また、はじめて通信する相手に対しては、何らかの方法で秘密鍵の配送が必要であるという欠点がある。この問題を避けるために、大規模なネットワークでは鍵管理センタを設置し、センタとの間の秘密鍵のみを保管し、暗号通信を行う場合はセンタから送信相手との秘密鍵を得る方法が用いられる。この場合秘密鍵の総数はｎとなる。

【０００７】一方、公開鍵暗号系では、個人が秘密に保管する鍵は自分の秘密鍵のみであり、必要な総秘密件数もｎ人のネットワークの場合ｎ個である。また、はじめて通信する相手に対しては、公開鍵の配送を行えばよく、鍵管理センタを設置して、ユーザの公開鍵をｎ個公開簿に登録し、センタから送信相手の公開鍵を得る方法が用いられる。この場合、センタは公開鍵の改ざんを防ぐだけで、秘密に保管する必要がない。但し、公開鍵方式は秘密鍵方式に比べて鍵のビット数が大きいため保管に要するファイルサイズは大きくなる。

【０００８】また、認証の場合、秘密暗号系では、例えば送信するメッセージを秘密鍵で圧縮変換し、送信文に付加して送り、受信側では同様に圧縮変換して比較する方式がとられている。しかし、送受信が同じ鍵であるため受信者は認証データを偽造することができる。これに対して、公開鍵暗号系では、秘密鍵で暗号化することができるのは本人だけであるという特徴を利用する。送信者はメッセージを圧縮変換して秘密鍵で暗号化し、送信文に付加して送り、受信者は送信者の公開鍵で付加されたデータを復号し、同様に圧縮変換したものと比較する方式がとられている。この場合は受信者は不正ができない。

【０００９】このように、認証系では公開鍵暗号系の技術は必要不可欠であるといえる。しかし、公開鍵暗号系には、暗号化／復号に大量の処理が必要であるという大きな欠点があるため、一般には処理の速い秘密暗号系をメッセージの暗号化に、公開鍵暗号系は認証用にというように組み合わせて用いられる場合が多い。公開暗号系の中で、現在最も有力なものが１９７７年にリヴェスト(Rivest)、シャミア(Shamir)およびエイドルマン(Adlman)の３人によって発明されたＲＳＡ暗号である。ＲＳＡ暗号の基本原理は次のようなものである。

【００１０】〈ＲＳＡの基本アルゴリズム〉暗号鍵（ｅ，Ｎ）と対応する復号鍵（ｄ，Ｎ）で、ｅとＮは公開鍵であり、ｄは秘密鍵である。平文をＭ、暗号文をＣとすると、暗号化Ｅと複合化Ｄのアルゴリズムは次のようにあらわされる。

$C=E(M)=M^e \bmod N$
$M=D(C)=C^d \bmod N$

但し、$d \cdot e=1 \bmod LCM\{(p-1),(q-1)\}$
$N=p \cdot q$
ＬＣＭ：最小公倍数[lowest common multiple]；ｐ、ｑは大きな素数
である。

【００１１】通常、ｅ、ｄ、Ｎ、Ｍなどは５１２ビット程度の大きな整数が用いられるので、高速指数計算法を使用しても１回のＲＳＡ演算で平均７７０回程度の乗算と剰余演算を行わなければならない。特に剰余演算は、近似法や剰余テーブル方式、モンゴメリのアルゴリズム等、多くの高速化手法が提案されている。このような、ＲＳＡ暗号に代表される公開鍵暗号系の多くで利用される、べき乗剰余アルゴリズムを高速に処理するためには、１回あたりの剰余アルゴリズムの高速化が要求される。

【００１２】この剰余演算の高速化の実現の一方法であるモンゴメリのアルゴリズムについて説明する。

〈モンゴメリのアルゴリズム〉モンゴメリのアルゴリズムは、剰余の法Ｎ（Ｎ＞１）と、剰余の法Ｎと互いに素である基数Ｒ（Ｒ＞Ｎ）を用いると、被剰余数ＴからＴ$R^{-1} \bmod N$の計算が基数Ｒによる除算のみで行えることを利用して、Ｎによる除算を用いることなく剰余計算を行うアルゴリズムである。ここで、Ｎ、Ｎ′、Ｒ、R^{-1}およびＴは整数であり、被剰余数Ｔは$0 \leqq T < RN$、R^{-1}は剰余の法Ｎの上での基数Ｒの逆数であり、$RR^{-1}-NN'=1$（$0 \leqq R^{-1} < N$，$0 \leqq N' < R$）の関係を満たす。

【００１３】さらにこの基数Ｒに２のベキ乗数を使用した場合、基数Ｒによる除算をシフト操作に置き換えることができるため、Ｔ→Ｔ$R^{-1} \bmod N$の計算の高速処理が可能となる。次にアルゴリズム１として、Ｔ→Ｔ$R^{-1} \bmod N$のアルゴリズムＲＥＤＣ（Ｔ）を示す。但し、アルゴリズム１において（Ｔ＋ｍＮ）／Ｒは必ず割り切れる。

【００１４】〈アルゴリズム１〉Ｔ→Ｔ$R^{-1} \bmod N$のアルゴリズムＲＥＤＣ（Ｔ）は次のようにあらわされる。

$m=(T \bmod R)\,N' \bmod R$
$t=(T+mN)/R$
if $t<N$ then return t else return $t-N$

すなわち、

$REDC(T)=t \quad (t<N)$
$=t-N \quad (t \geqq N)$

である。

【００１５】１回のＲＥＤＣでは、剰余$T \bmod N$ではなくＴ$R^{-1} \bmod N$が求められるだけである。そこで、剰余$T \bmod N$を求めるためには、次に示すようにＲＥＤＣ（Ｔ）と予め求めておいた$R^2 \bmod N$との積で、再びＲＥＤＣを行えばよい。

$REDC(REDC(T)*(R^2 \bmod N))$
$=(TR^{-1} \bmod N)(R^2 \bmod N)R^{-1} \bmod N$
$=TR^{-1}*R^2*R^{-1} \bmod N$
$=T \bmod N$

このようにして、剰余$T \bmod N$を求めることができる。

【００１６】〈ＲＥＤＣの多重精度計算への拡張〉次に、剰余の法Ｎあるいは基数Ｒが多倍長すなわち多重精度の場合について、ＲＥＤＣのアルゴリズムを拡張する。剰余の法Ｎや基数Ｒが多重精度の場合、ＲＥＤＣの（$T \bmod R$）Ｎ′やｍＮの計算は、多重精度×多重精度の処理となり汎用の計算機では非常に大きな処理量と処理時間が必要となる。そこで、この部分を多重精度×単精度の処理で行えるように拡張したアルゴリズム２を示す。

【００１７】〈アルゴリズム２〉ＲＥＤＣを多重精度へ拡張したアルゴリズムは次に示すようになる。被剰余数Ｔがｂ進数で$T=(T_{2n-1}T_{2n-2}\cdots T_0)_b$、$R=b^n$、$b=2^k$とあらわされる場合、次に示すｉ＝０～ｎ－１の繰り返し処理により$TR^{-1} \bmod N$を単倍長と同様にして求めることができる。

【００１８】

```
for i=0 to n-1
    m′=T0N′ mod b
    T=T+m′N
    t=T/b
next
if t<N then return t else return t-N
```

このようにして得られる$TR^{-1} \bmod N$と、上述したように予め求めておいた$R^2 \bmod N$との積で再びＲＥＤＣを行うことにより求める剰余$T \bmod N$を求めることができる。

【００１９】

【発明が解決しようとする課題】上述したモンゴメリのアルゴリズムは、基数Ｒを２のべき乗数にすることによ

り、処理時間のかかる割り算を高速なシフト処理に置き換えることが可能であるため剰余計算を高速で行うことができる。しかし、このモンゴメリのアルゴリズムを用いた演算を高速で行おうとすると、多重精度の処理は、多重精度×単精度の乗算処理が必要となるため、大量の処理が必要となる。

【００２０】また、べき乗剰余などの演算をこのアルゴリズムを用いて行う場合、剰余演算が何度も繰り返されることになるため、剰余演算の処理回数の多い部分をできるだけ単純化することで全体の計算量を削減し、処理速度の向上を図る必要がある。また、モンゴメリ法特有のN′ およびR²modN等の事前に計算しておかなければならないパラメータの計算が前処理として必要である。特に、N′ は、拡張ユークリッドの互除法を使用するため、R²modNよりも多くの処理時間を必要とする。

【００２１】したがって、本発明は、剰余計算に際し多重の繰り返し演算を必要とする倍数演算部分に着目して剰余計算の演算処理を簡略化し、高速処理を実現し得る剰余計算装置および乗算剰余計算装置を提供することを目的としている。本発明の他の目的は、本発明による剰余計算装置および乗算剰余計算装置に用いるための倍数テーブルを容易に作成することのできるテーブル作成装置を提供することにある。

【００２２】

【課題を解決するための手段】本発明は、上記目的を達成するために、多重の繰り返し演算を必要とする倍数演算部分にテーブルを用い、演算結果をテーブル検索により得るようにして、演算処理を簡略化して高速化することにより、剰余計算を高速処理し得る次のような構成の剰余計算装置とした。

【００２３】図１は、本発明の基本となる剰余計算装置の原理を示している。図１に示す剰余計算装置は、入力レジスタ１、倍数算定部２、加算器３および補正装置４を有している。入力レジスタ１は入力される被剰余数Tを保持する。倍数算定部２は、入力レジスタ１の下位所定ビットに対応して剰余の法Nの倍数が格納された倍数テーブル５を有し、入力レジスタ１の被剰余数Tの下位所定ビットに基づいて倍数テーブル５を検索することにより、前記Tの下位所定ビットに対応する剰余の法Nの倍数を求める。加算器３は、倍数算定部２で求められる剰余の法Nの倍数と入力レジスタ１の内容とを加算する。この加算器３の所定回数ｎの加算について、加算器３の加算結果の上位所定ビットで入力レジスタ１の内容を更新する。補正装置４は、加算器３の加算結果として得られるｔに対して補正処理を施す。

【００２４】このような剰余計算装置は、ｎ回の繰り返し演算の必要な倍数算定部２に倍数テーブル５を用いて高速に倍数を求めることができるので、剰余計算を高速に処理することができる。図１の剰余計算装置においては、具体的には、先に述べたモンゴメリのアルゴリズムにおける被剰余数Tが入力レジスタ１に格納され、倍数算定部２において剰余の法Nの倍数m′ N（但し、m′ ＝（Tmodb）N′ modb、ここでN′ およびRは整数であり、R＝bⁿ、b＝２ᵏとする）が格納された倍数テーブル５を検索して、前記被剰余数Tの下位ｋビットに対応する剰余の法Nの倍数m′ Nが求められる。この倍数m′ Nと前記被剰余数Tが加算され、その加算結果により入力レジスタ１の内容が更新される。このような処理がｎ回繰り返され、加算器３の出力にｔ＝（T＋mN）／Rが得られる。この加算器３の出力ｔは、補正装置４に与えられ、ｔ＜Nの場合はそのままｔ＝REDC（T）、ｔ≧Nの場合はｔ－N＝REDC（T）として、TR⁻¹modNが得られる。

【００２５】上述の倍数テーブル５を作成するテーブル作成装置は、繰り返し加算部、補数算定部およびテーブル書き込み部を有する。繰り返し加算部は剰余の法Nを繰り返し加算する。補数算定部は前記繰り返し加算部で求められる各加算結果の値の下位ｂビットの２の補数を求める。テーブル書き込み部は、テーブルの前記補数算定部で得られる各補数値に対応するテーブルアドレスに前記各加算結果をそれぞれ格納する。このテーブル作成装置は、例えば同様の機能をソフトウェアにより実現するコンピュータにより構成することもできる。

【００２６】このようなテーブル作成装置により、計算処理に多くの時間を必要とするN′ を求めることなく記憶装置上に倍数テーブルを作成することができる。また、上述の倍数テーブル５は、多量のデータにより構成されることになり、倍数テーブル５を構成する記憶装置に多くの記憶容量を必要とするので、倍数テーブル５を用いる倍数算定部２に代えて、被剰余数Tの下位所定ビットに対応させて倍数情報m′ を格納する倍数情報テーブルを用い、この倍数情報テーブルを前記被剰余数Tの下位所定ビットで検索して、それに対応する倍数情報m′ を求める倍数情報算定部と、剰余の法Nを格納する剰余の法レジスタと、前記倍数情報算定部で得られる倍数情報m′ と前記剰余の法レジスタから得られる剰余の法Nとを乗算する乗算器とを有する倍数算定部を設ける構成としてもよい。この場合、前記倍数算定部により前記被剰余数の下位所定ビットに対応する前記剰余の法Nの倍数m′ Nを求める。この場合、計算速度は、倍数テーブル５を用いる場合よりも遅くなるが、倍数情報テーブルは倍数テーブル５に比して少ない記憶容量で済む。

【００２７】さらに、乗算剰余計算装置は、上述した剰余計算装置と同様の構成に加えて、被乗数に対し所定ビット毎に乗数との乗算を行い行う分割乗算装置を設け、この分割乗算装置の出力を上述した剰余計算装置の入力レジスタに与えるとともに、前記剰余計算装置の加算器の出力を前記分割乗算装置の乗算結果に加算する構成として、乗算剰余の計算を高速に行う。

【００２８】

【作用】本発明による剰余計算装置では、剰余計算において、ループによる多数回の繰り返し演算が必要な倍数算定部２に、入力レジスタ１の保持値の下位所定ビットに基づくテーブル検索を用いて高速に処理することにより、高速剰余計算が行われる。特に、倍数算定部２に倍数テーブル５を用いて、倍数算定部２の機能をすべてテーブル検索により処理すれば、処理速度が顕著に向上する。

【００２９】本発明によるテーブル作成装置では、倍数算定部２に用いる倍数テーブル５を作成するのに、剰余の法を繰り返し加算し、加算結果の下位所定ビットの２の補数を求め、その２の補数値に対応するアドレスに前記加算結果を格納することにより、煩雑な計算を要するパラメータN′を求めることなく倍数テーブル５を作成することができる。

【００３０】本発明による乗算剰余計算装置では、剰余計算における繰り返し処理の部分にテーブルを用いて、効率よく剰余計算処理を高速化し、乗算剰余計算を処理速度を向上させる。

【００３１】

【実施例】

〈実施例１〉本発明に係る剰余計算装置の第１の実施例の説明に先立ち、本発明における剰余計算処理の原理を詳細に説明する。上述したモンゴメリのアルゴリズムを利用して、ハードウェアにより剰余計算装置を構成すると、一般的には図８のようになると考えられる。図８の構成は、先に述べた多重精度のアルゴリズムをそのままハードウェアで実現したものである。

【００３２】図８の剰余計算装置は、先に述べた通り、$R=b^n$、$b=2^k$として、$T \rightarrow TR^{-1} \bmod N$の処理を行うものとし、Ｔレジスタ１１、Ｎレジスタ１２、N′レジスタ１３、第１の乗算器１４、第２の乗算器１５、第１の加算器１６、第１のレジスタ１７、－Ｎレジスタ１８、第２の加算器１９、第２のレジスタ２０およびセレクタ２１を具備する。

【００３３】第１のレジスタ１７、－Ｎレジスタ１８、第２の加算器１９、第２のレジスタ２０およびセレクタ２１は、補正装置２２を構成する。N′レジスタ１３および第２の乗算器１５からなる部分は倍数情報算定部３１を構成し、この倍数情報算定部３１、Ｎレジスタ１２および第１の乗算器１４からなる部分は倍数算定部３２を構成する。

【００３４】ここで、剰余の法Ｎをａビット、入力値をａ＋ｎｋビットと仮定して、上述の剰余計算装置について説明する。Ｔレジスタ１１は、図１における入力レジスタ１に相当し、入力される被剰余数Ｔを保持する。Ｎレジスタ１２およびN′レジスタ１３は、共にａビットのレジスタであり、与えられる剰余の法Ｎおよび先に述べたN′をそれぞれ保持する。第２の乗算器１５は、Ｔレジスタ１１の保持値の下位ｋビットとN′レジスタ１３の下位ｋビットとを乗算する。第１の乗算器１４は、Ｎレジスタ１２の保持値と第２の乗算器１５の乗算結果の下位ｋビットとを乗算する。

【００３５】第１の加算器１６は、Ｔレジスタ１１の保持値（ａ＋ｎｋビット）と第１の乗算器１４の乗算結果（ａ＋ｋビット）とを加算する。第１の加算器１６の加算結果（ａ＋ｎｋ＋１ビット）の上位ａ＋（ｎ－１）ｋ＋１ビットはＴレジスタ１１の内容を更新してｎ回のループ処理を行うのに用いられる。第１のレジスタ１７は、第１の加算器１６の加算結果の下位ｋビットを超える部分の下位ａ＋１ビットを保持し、－Ｎレジスタ１８は、予め与えられる剰余の法Ｎに基づく－Ｎの値（ａ＋１ビット）を保持する。第２の加算器１９は、第１のレジスタ１７の保持値と－Ｎレジスタ１８の保持値とを加算して、第２のレジスタ２０に保持させる。セレクタ２１は、第２のレジスタ２０のａ＋１ビットの保持値の最上位の１ビット（符号ビットに相当する）の値を条件として動作し、該最上位ビットが“１”の場合は第１のレジスタ１７の保持値を出力し、“０”の場合は第２のレジスタ２０の保持値を出力する。

【００３６】すなわち、剰余計算を行う場合、前処理として事前に計算して求めておいたＮ、N′および－Ｎを、Ｎレジスタ、N′レジスタおよび－Ｎレジスタにそれぞれ設定する。入力被剰余数Ｔは、まずＴレジスタ１１に与えられる。このＴレジスタ１１の保持値の下位ｋビットとN′レジスタ１３の保持値の下位ｋビットとを第２の乗算器１５に入力する。第２の乗算器１５の乗算結果の下位ｋビットとＮレジスタ１２の保持値を第１の乗算器１４に入力し、その乗算結果とＴレジスタ１１の保持値とを第１の加算器１６に入力する。Ｔレジスタ１１と第１の加算器１６との間のループ処理の回数がｎ未満の場合は、第１の加算器１６の加算結果の上位ａ＋（ｎ－１）ｋ＋１ビットを再びＴレジスタ１１に入力する。ループ処理の回数がｎに達した場合は、第１の加算器１６の加算結果の下位ｋ＋１ビット目から下位ａ＋ｋ＋２ビット目までのａ＋１ビットを取り出し、補正装置２２の第１のレジスタ１７に入力する。

【００３７】補正装置２２では、入力を一旦第１のレジスタ１７に保持し、その保持値と－Ｎレジスタ１８の保持値とを第２の加算器１９に入力し、加算結果を第２のレジスタ２０に与える。第２のレジスタ２０の上位１ビットの値をセレクタ２１の動作条件として、“１”の場合は第１のレジスタ１７の保持値が、“０”の場合は第２のレジスタ２０の保持値がセレクタ２１で選択され、結果として出力される。

【００３８】ここで、補正装置２２における処理は、モンゴメリのアルゴリズムで最後に計算値ｔがＮ以上の場合にはＮを引く部分の処理である。この図８のような構成の場合、倍数算定部３２の部分は何度も同じ処理を繰り返されることになる。さらにこの構成をべき乗剰余演

算に用いた場合はこれらの処理全体が繰り返されるため、倍数算定部３２の部分の処理回数は非常に多くなる。

【００３９】一方、倍数算定部３２の処理自体は、乗算という処理量の大きな演算が中心であり、さらに計算内容がＮという多重精度の定数とｋビットの変数との積であるため処理回数の多さとあいまって、長い処理時間が必要となる。そこで、本発明の第１の実施例では、図８における倍数情報算定部３１の部分をｋビット×2^kの大きさの倍数情報テーブルとして持っておくことにより、無駄な計算を省き、処理を高速化する。

【００４０】本発明に係る剰余計算装置の第１の実施例を説明する。なお、以下の各実施例においては、説明の便宜のためビット数をａ＝５１２、ｋ＝８、ｎ＝６４として説明する。図２は、倍数情報テーブル方式を用いた剰余計算装置の構成を示している。図２において、図８と共通の部分には同符号を付してその詳細な説明を省略する。図２の剰余計算装置は、図８と同様の、Ｔレジスタ１１、Ｎレジスタ１２、第１の乗算器１４、第１の加算器１６および補正装置２２を具備する。補正装置２２は、第１のレジスタ１７、－Ｎレジスタ１８、第２の加算器１９、第２のレジスタ２０およびセレクタ２１を有する。

【００４１】さらに、図２の剰余計算装置においては、図８における倍数情報算定部３１に代えて、倍数情報テーブル４１を備えた倍数情報算定部４２を設けている。倍数情報テーブル４１は、被剰余数Ｔの下位所定ビットに対応して倍数情報m′を格納している。Ｔレジスタ１１は、１０２４ビットの入力レジスタであり、このＴレジスタ１１には、被剰余数Ｔおよび第１の加算器１６の出力ループが再び入力される。倍数情報テーブル４１は、幅８ビット奥行き２５６のテーブルで、Ｔレジスタ１１の下位８ビットで索引すると８ビットの倍数情報m′を出力する。Ｎレジスタ１２は、剰余の法Ｎを格納する５１２ビットのレジスタである。

【００４２】第１の乗算器１４は、８ビット×５１２ビットの乗算器であり、倍数情報テーブル４１の索引結果とＮレジスタ１２の保持値との乗算を行う。第１の加算器１６は、１０２５ビットの加算器であり、Ｔレジスタ１１の保持値と第１の乗算器１４の乗算結果との加算を行う。補正装置２２は、図８の補正装置２２と同じもので剰余計算結果の補正を行う。

【００４３】倍数情報算定部４２の倍数情報テーブル４１は、図３に示すように予めＮ′の下位８ビットを加算器により２５５回順次加算し、結果の下位８ビットを加算回数をアドレスとしてテーブル化したものである。被剰余数ＴがＴレジスタ１１に入力されると、その下位８ビットで倍数情報テーブル４１の索引を行い、その索引結果（＝m′）とＮレジスタ１２の保持値Ｎを第１の乗算器１４で乗算し、その積とＴレジスタ１１の保持値を第１の加算器１６で加算する。ループの回数ｎが６４未満の場合は、第１の加算器１６の加算結果の上位１０１８ビットを、再びＴレジスタ１１にＬＳＢ（leastsignificant bit）側からつめて入力し、処理ループを形成する。ループの回数ｎが６４回目の場合は、第１の加算器１６の加算結果の下位９ビット目から５１３ビットを取り出して、補正装置２２の第１のレジスタ１７に入力する。補正処理は図８の場合と同様である。

【００４４】このように、図２の剰余計算装置とすれば、倍数情報の算定を倍数情報算定部４２でテーブル索引により行うので、倍数情報の算定を高速に行うことができ、剰余計算の高速化を実現することができる。なお、上述のように、図８における倍数情報算定部３１の部分をテーブル化するよりも、図８における倍数算定部３２の部分をテーブル化したほうが、当然高速化されるが、テーブルのサイズが大きくなるため、記憶装置に大きなテーブルをとる余裕がない場合に倍数情報算定部３１の部分のテーブル化を行うことにより、小規模の装置構成でも高速化を実現することができる。

【００４５】次に、図８における倍数算定部３２の部分をテーブル化する場合の実施例を説明する。

〈実施例２〉本発明の第２の実施例では、図８における倍数算定部３２の部分をａ＋ｋビット×2^kの大きさの倍数テーブルとして持っておくことにより、無駄な計算を省き、処理を高速化する。以下、本発明の第２の実施例に係る剰余計算装置を、図２または図８と同様の部分には同符号を付して示す図４を参照して説明する。

【００４６】図４に示す倍数テーブル方式を用いた剰余計算装置は、図２と同様の、Ｔレジスタ１１、第１の加算器１６および補正装置２２を具備する。さらに、図４の剰余計算装置においては、図８における倍数算定部３２に代えて、倍数テーブル５１を備えた倍数算定部５２を設ける。倍数算定部５２の倍数テーブル５１は、被剰余数Ｔの下位所定ビットに対応させて剰余の法Ｎの倍数m′Ｎを格納している。すなわち、倍数テーブル５１は、図２の倍数情報テーブル４１、Ｎレジスタ１２および第１の乗算器１４からなる部分の処理を一括してテーブル化したものであり、Ｔレジスタ１１の保持値の下位８ビットで検索すると５２０ビットの倍数算定結果が出力される。この倍数テーブル５１は、例えば後述する第３の実施例のような倍数テーブル作成方法に従って、予め幅５２０ビット奥行き２５６のテーブルとして作成する。

【００４７】第１の実施例の場合と同様に被剰余数ＴがＴレジスタ１１に入力されると、Ｔレジスタ１１の下位８ビットで倍数算定部５２の倍数テーブル５１を索引し、その結果とＴレジスタ１１の値を第１の加算器１６で加算する。第１の加算器１６の加算後の処理、つまりＴレジスタ１１と第１の加算器１６との間の加算ループ処理以後の処理は上述した第１の実施例と同様に行われ

る。

【００４８】このように、図４の剰余計算装置とすれば、倍数算定を倍数算定部５２で、場数テーブル５１のテーブル索引により行うので、倍数の算定を一層高速に行うことができる。

〈実施例３〉本発明の第３の実施例は、上述した第２の実施例による図４に示した剰余計算装置で用いるための倍数テーブルを作成するためのテーブル作成装置であり、倍数テーブルを容易に作成することができる。

【００４９】本発明の第３の実施例のテーブル作成装置は、モンゴメリのアルゴリズムの剰余の法Ｎの倍数を入力値Ｔに加えると、下位ｋビットが０となるという性質を利用し、パラメータＮ′を直接使用することなく倍数テーブルの作成を行う。図５にテーブル作成装置の構成および作成される倍数テーブルの構成を模式的に示している。

【００５０】テーブル作成装置は、Ｎレジスタ６１、加算器６２、加算結果レジスタ６３および補数演算部６４を有する。Ｎレジスタ６１は、剰余の法Ｎを保持し、このＮレジスタ６１の保持値を加算器６２で繰り返し加算する。この加算器６２の加算結果は加算結果レジスタ６３に保持され、補数演算部６４は、加算結果レジスタ６３の保持値の下位所定ビットの２の補数を求める。このような計算の結果、逐次、補数演算部６４で求められた値をテーブルのアドレスとして、加算結果レジスタ６３に保持された値を書き込んで、倍数テーブル５１とする。

【００５１】ここでは、説明の便宜のため、剰余の法Ｎを10110011_2とし、索引を４ビットで行うものとする。図５において、まず、０をテーブルの０番目すなわちアドレス０にセットする。次に０にＮレジスタ６１の剰余の法Ｎを加え、加算結果を10110011_2として、加算結果レジスタ６３に保持する。この加算結果の下位４ビットを取り出し、補数演算部６４で２の補数をとると1101_2になるので、テーブルのアドレス1101_2に10110011_2をセットする。このようにして、順次、テーブルの求められた２の補数値のアドレスに加算結果をセットしてゆく。

【００５２】例えば、図５に示された状態では、剰余の法Ｎを６回加算した値100001100102_2に、Ｎを加えて100111001012_2を得て、下位４ビットの２の補数1011_2をとる。そして、テーブルのアドレス1011_2に100111001012_2をセットしている。このような操作を１６回繰り返すことにより、倍数テーブル５１を作成することができる。

【００５３】この実施例によるテーブル作成装置は、煩雑な計算を要するパラメータＮ′を直接使用することなく、モンゴメリのアルゴリズムの剰余の法Ｎの倍数を入力値Ｔに加えると、下位ｋビットが０となるという性質を利用して、倍数テーブルを容易に作成することができる。なお、このテーブル作成装置は、単純な繰り返し加算処理、２の補数を求める処理およびテーブルの書き込み処理のみで済むので、図５に対応するハードウェア構成のみならず、ワークステーション、パーソナルコンピュータ等を用いたソフトウェア処理で容易に実現することができる。

【００５４】〈実施例４〉本発明の第４の実施例は、図２に示したように倍数情報テーブルを用いた剰余計算装置の構成を、分割乗算処理を行う分割乗算装置と組み合わせて構成した乗算剰余計算装置の実施例である。以下、本発明の第４の実施例に係る乗算剰余計算装置を、図２と同様の部分には同符号を付して示す図６を参照して説明する。

【００５５】図６に示す乗算剰余計算装置は、Ｒ＝b^n、$b=2^k$として、$AB \rightarrow ABR^{-1} \bmod N$の処理を行うものとし、図２と同様の、Ｔレジスタ１１、Ｎレジスタ１２、第１の乗算器１４、第１の加算器１６、補正装置２２および倍数情報テーブル４１を具備する。さらに、図６の乗算剰余計算装置においては、Ａレジスタ７１、Ｂレジスタ７２、第３の乗算器７３および第３の加算器７４を設けている。

【００５６】Ａレジスタ７１、Ｂレジスタ７２、第３の乗算器７３および第３の加算器７４は分割乗算装置７５を構成しており、Ｔレジスタ１１、Ｎレジスタ１２、第１の乗算器１４、第１の加算器１６および倍数情報テーブル４１は剰余演算装置７６を構成している。Ａレジスタ７１は与えられる乗数Ａを保持し、Ｂレジスタ７２は与えられる被乗数Ｂを保持する。乗算器７３は、Ｂレジスタ７２に保持される被乗数をｋビット毎に下位から順次取り出して、Ａレジスタ７１の乗数Ａに乗算する。加算器７４は、ｎ回のループ処理の間、剰余演算装置７６の加算器１６の演算結果を乗算器７３の乗算結果に加算する。

【００５７】この場合、図２の場合とは異なり、剰余演算装置７６の加算器１６の加算結果はそのまま加算器７４に入力されてｎ回の処理ループを形成する。ここで、図６の乗算剰余計算装置は、図２におけるＴレジスタ１１と処理ループの部分に、５１２ビット幅のＡレジスタ７１およびＢレジスタ７２、５２０ビット幅の乗算器７３、ならびに５２１ビット幅の加算器７４を具備する分割乗算装置７５を組み込んだ倍数情報テーブル式の乗算剰余計算装置として、具体的なビット数の例を示して説明する。図２の剰余計算装置をそのまま乗算剰余計算装置に使用する場合は、乗算装置を設けて単にその乗算結果をＴレジスタ１１に代入するようにして実現することができるが、本実施例では、分割乗算装置と組み合わせることにより、剰余演算装置のＴレジスタ１１、加算器１６およびループ処理部分のビット幅を削減している。

【００５８】乗算剰余を行う２数すなわち乗数および被乗数が、ＡおよびＢのレジスタ７１および７２にそれぞ

れ入力されると、Ａレジスタ７１の保持値とＢレジスタ７２の保持値の下位側から８ビットずつの値が乗算器７３で乗算される。この乗算器７３の乗算結果と前回の処理ループの計算値（初回は０とする）が加算器７４で加算されてＴレジスタ１１に入力される。ここから加算器１６までは図２の実施例の場合と同様の処理が行われる。ループの回数ｎが６４未満の場合、加算器１６の加算結果の上位５１４ビットをループによりフィードバックして加算器３に入力し、次回のＡレジスタ７１とＢレジスタ７２との分割剰余結果に加算する。ループの回数ｎが６４の場合、加算器１６の加算結果の下位９ビット目から５１３ビットを取り出し、補正装置２２に入力する。このようにすることにより、５１２ビットのＡＢＲ$^{-1}$modＮが補正装置２２から出力される。

【００５９】〈実施例５〉本発明の第５の実施例は、図４に示したように倍数テーブルを用いた剰余計算装置の構成を、分割乗算処理を行う分割乗算装置と組み合わせて構成した乗算剰余計算装置の実施例である。以下、本発明の第５の実施例に係る乗算剰余計算装置を、図４または図６と同様の部分には同符号を付して示す図７を参照して説明する。

【００６０】図７に示す乗算剰余計算装置は、図４と同様の、Ｔレジスタ１１、第１の加算器１６、補正装置２２および倍数テーブル５１を具備する。さらに、図７の乗算剰余計算装置においては、図６と同様の、Ａレジスタ７１、Ｂレジスタ７２、第３の乗算器７３および第３の加算器７４を設けている。Ａレジスタ７１、Ｂレジスタ７２、第３の乗算器７３および第３の加算器７４は分割乗算装置７５を構成しており、Ｔレジスタ１１、第１の加算器１６および倍数テーブル５１は剰余演算装置８１を構成している。

【００６１】図６の場合と同様に、Ａレジスタ７１は与えられる乗数Ａを保持し、Ｂレジスタ７２は与えられる被乗数Ｂを保持する。乗算器７３は、Ｂレジスタ７２に保持される被乗数をｋビット毎に下位から順次取り出して、Ａレジスタ７１の乗数Ａに乗算する。加算器７４は、ｎ回のループ処理の間、剰余演算装置８１の加算器１６の演算結果を乗算器７３の乗算結果に加算する。加算器１６の加算結果はそのまま加算器７４に入力されてｎ回の処理ループを形成する。

【００６２】なお、本発明の剰余計算装置および乗算剰余計算装置はハードウェアに限らず同様の機能構成の少なくとも一部をソフトウェアで実現することもでき、そのような場合にも処理の高速化を達成することができる。

【００６３】

【発明の効果】以上説明したように、本発明によれば、モンゴメリ法による剰余計算処理における倍数算定の乗算処理をテーブル索引で行うため、処理を簡略化することができ、高速に剰余演算を行うことを可能とする剰余計算装置および同様に高速に乗算剰余演算を行うことを可能とする乗算剰余計算装置を提供することができる。

【００６４】例えば、ソフトウェアで本発明をの剰余計算装置または乗算剰余計算装置を実現した場合、剰余計算部分を比較すれば、テーブルを用いない場合の処理時間を１００とすると、倍数情報テーブル方式が９８、倍数テーブル方式が３１となり、その効果は明らかである。また、本発明によれば、倍数テーブル方式に用いる倍数テーブルを作成するのに、剰余の法を繰り返し加算し、加算結果の下位所定ビットの２の補数を求め、その２の補数値に対応するアドレスに前記加算結果を格納することにより、煩雑な計算を要するパラメータＮ′を求めることなくテーブルを作成することが可能なテーブル作成装置を提供することができる。

【図面の簡単な説明】

【図１】本発明に係る剰余計算装置の構成を示す原理図である。

【図２】本発明の第１の実施例に係る剰余計算装置の構成を示すブロック図である。

【図３】図２の剰余計算装置で用いる倍数情報テーブルを説明するための模式図である。

【図４】本発明の第２の実施例に係る剰余計算装置の構成を示すブロック図である。

【図５】図４の剰余計算装置で用いる倍数テーブルを作成するための本発明の第３の実施例に係るテーブル作成装置の原理構成を概略的に示す図である。

【図６】本発明の第４の実施例に係る乗算剰余計算装置の構成を示すブロック図である。

【図７】本発明の第５の実施例に係る乗算剰余計算装置の構成を示すブロック図である。

【図８】モンゴメリのアルゴリズムを単純にハードウェア化した場合の一般的に考えられる剰余計算装置の構成を示すブロック図である。

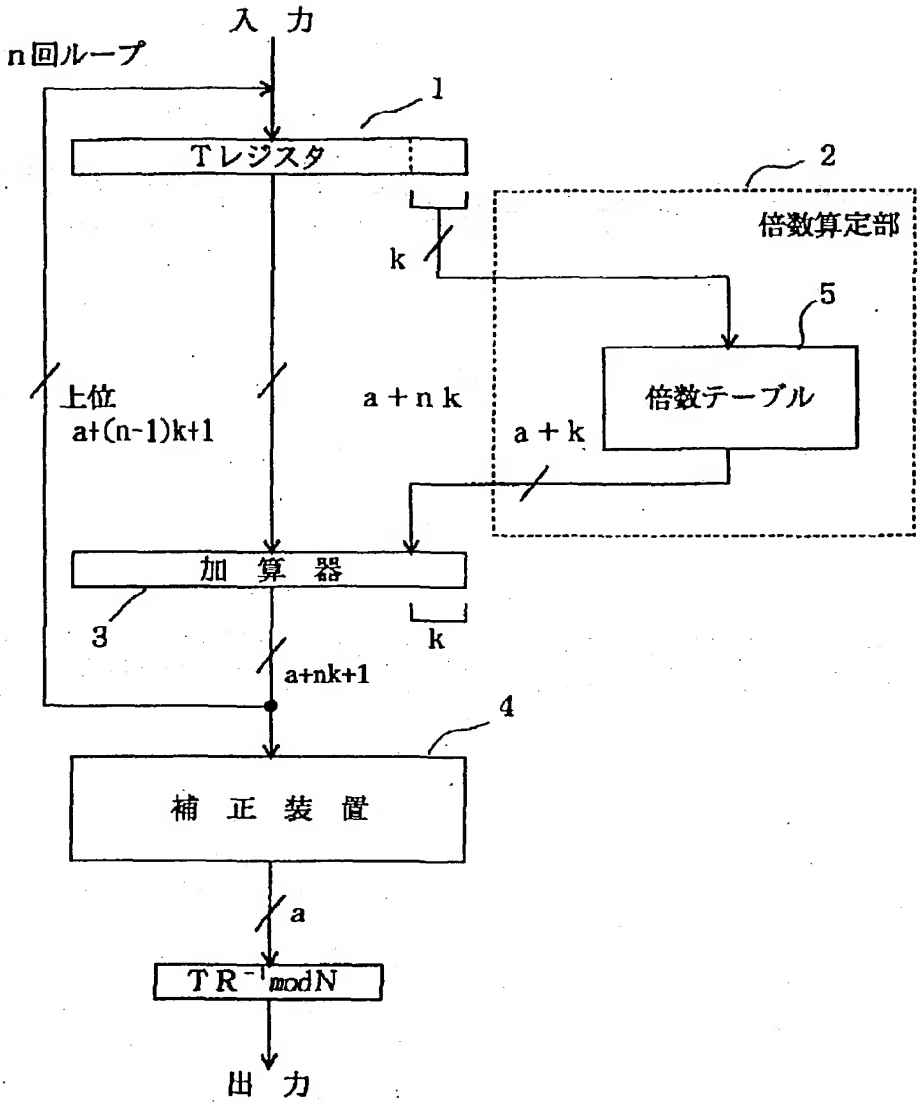
【符号の説明】

１，１１…Ｔレジスタ
２，５２…倍数算定部
３，１６，１９，６２，７４…加算器
４，２２…補正装置
５，５１…倍数テーブル
１２，６１…Ｎレジスタ
１４，７３…乗算器
１７，２０…レジスタ
１８…－Ｎレジスタ
２１…セレクタ
４１…倍数情報テーブル
４２…倍数情報算定部
６３…加算結果レジスタ
６４…補数演算部
７１…Ａレジスタ
７２…Ｂレジスタ

７５…分割乗算装置　　７６，８１…剰余演算装置

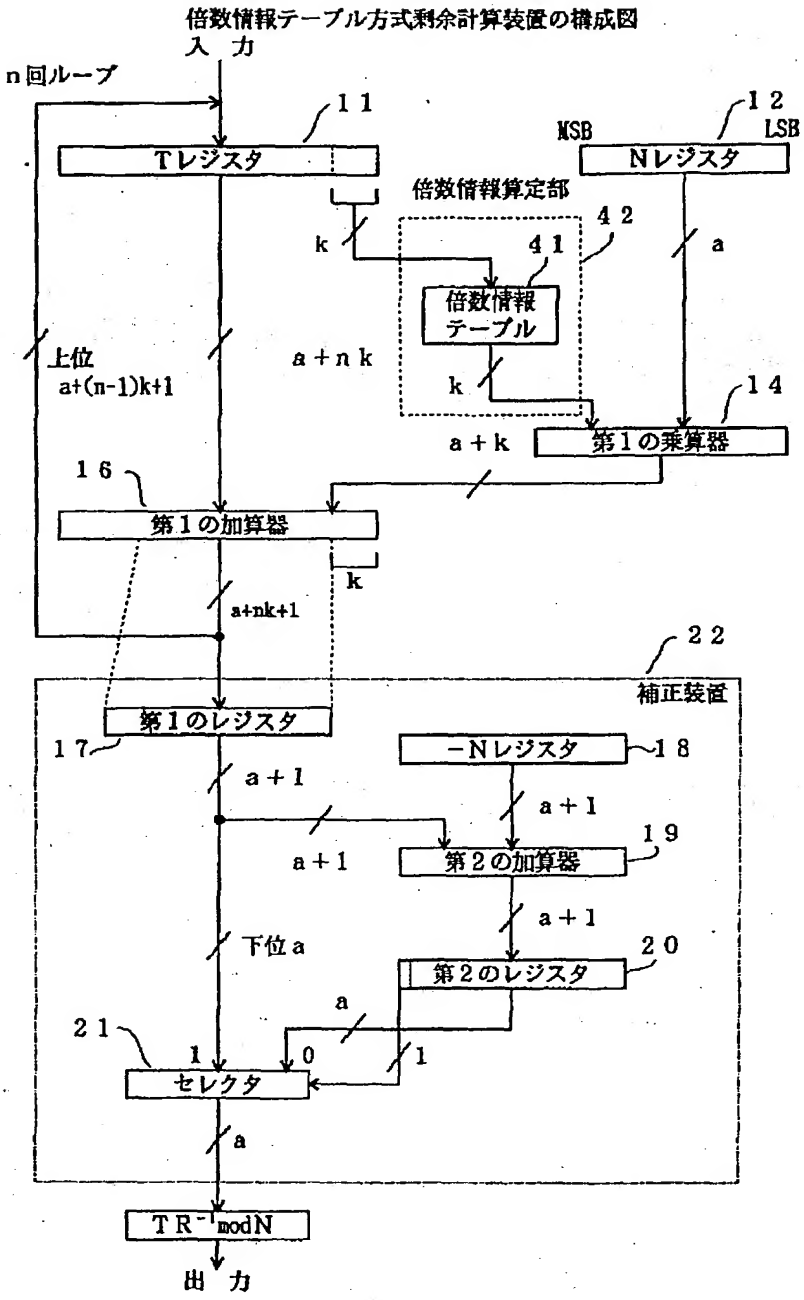
【図１】

剰余計算装置の原理図



$T \rightarrow TR^{-1} \bmod N \qquad R = b^n \qquad b = 2^k$

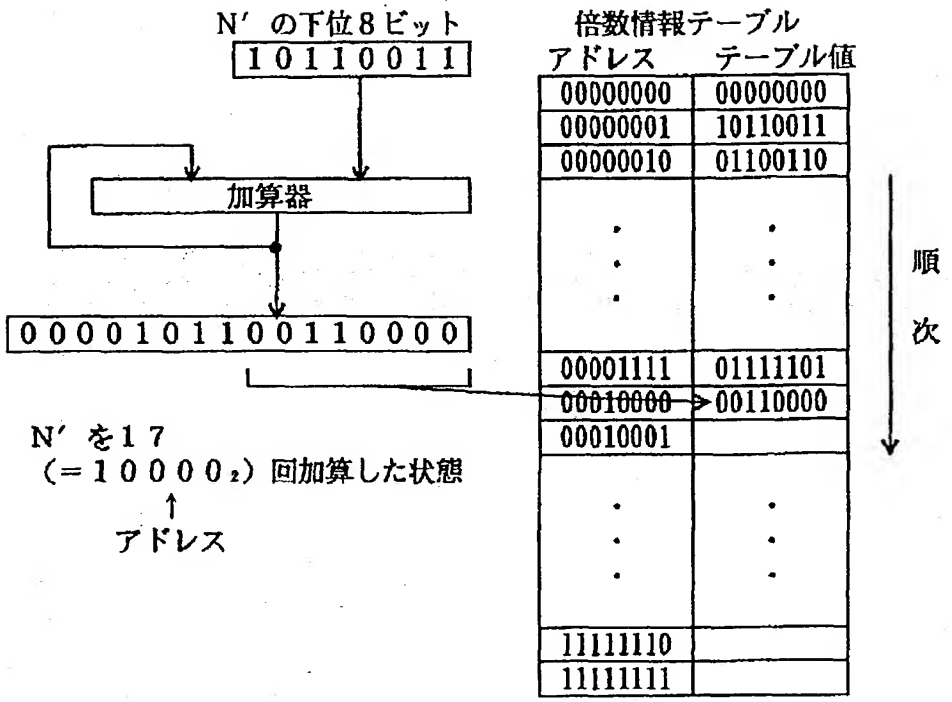
【図２】



$T \rightarrow TR^{-1} \bmod N \qquad R = b^n \qquad b = 2^k$

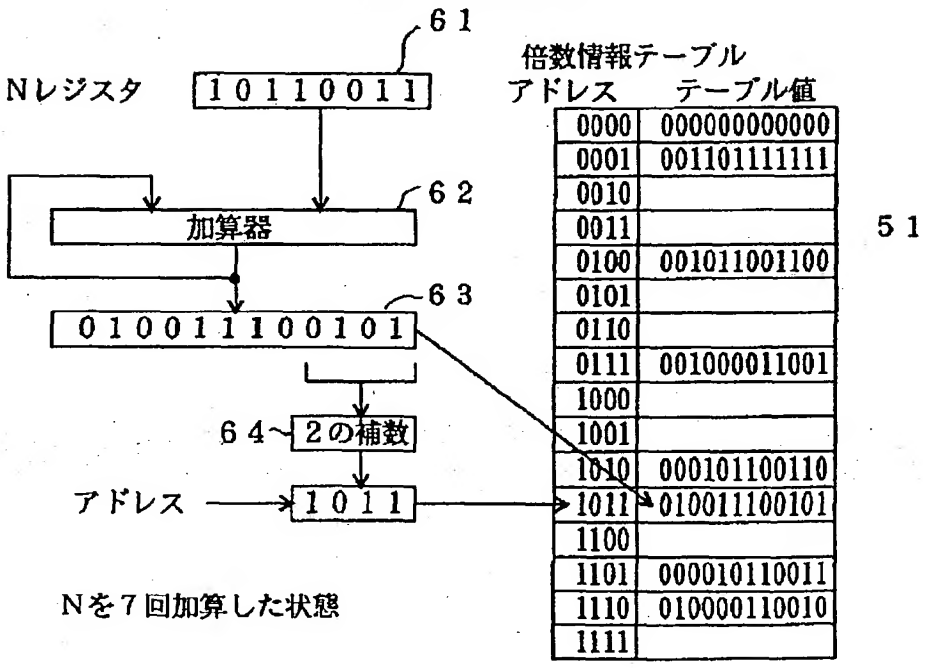
【図３】

倍数情報テーブルの作成

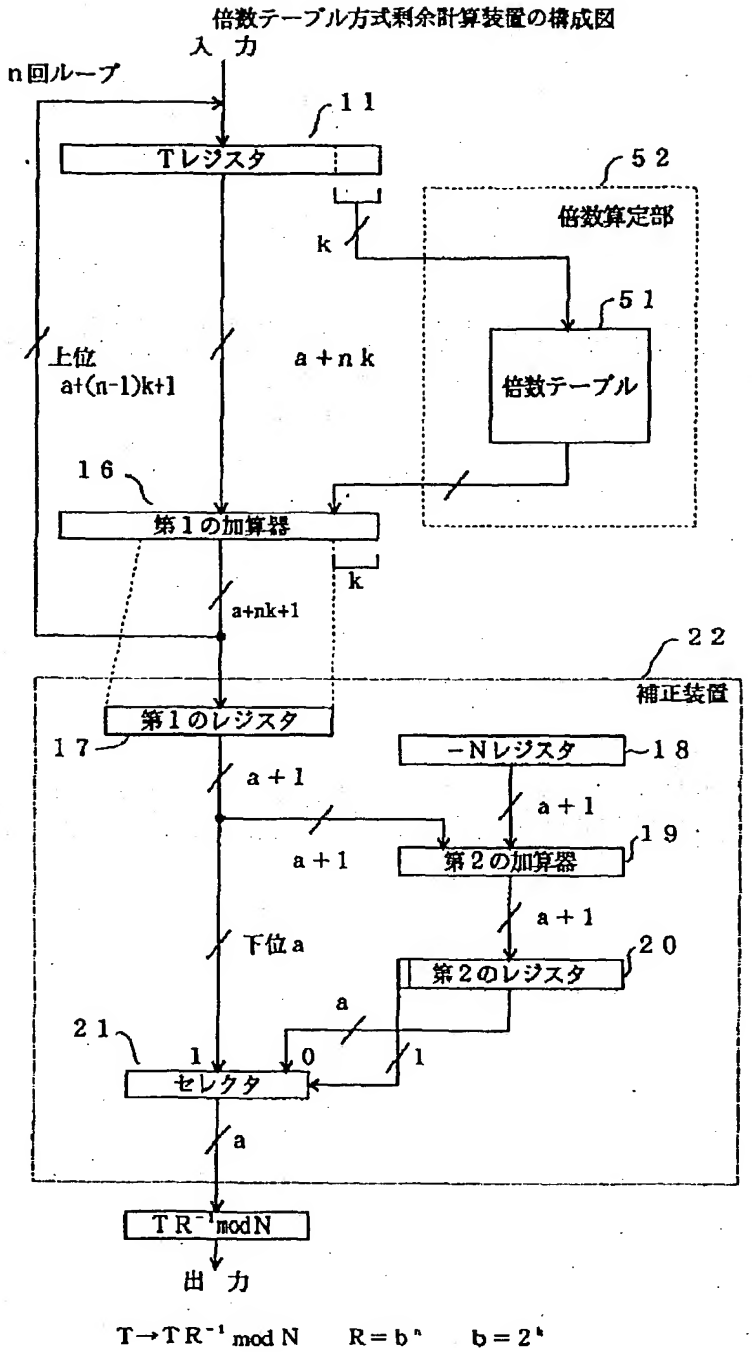


【図５】

倍数テーブルの作成

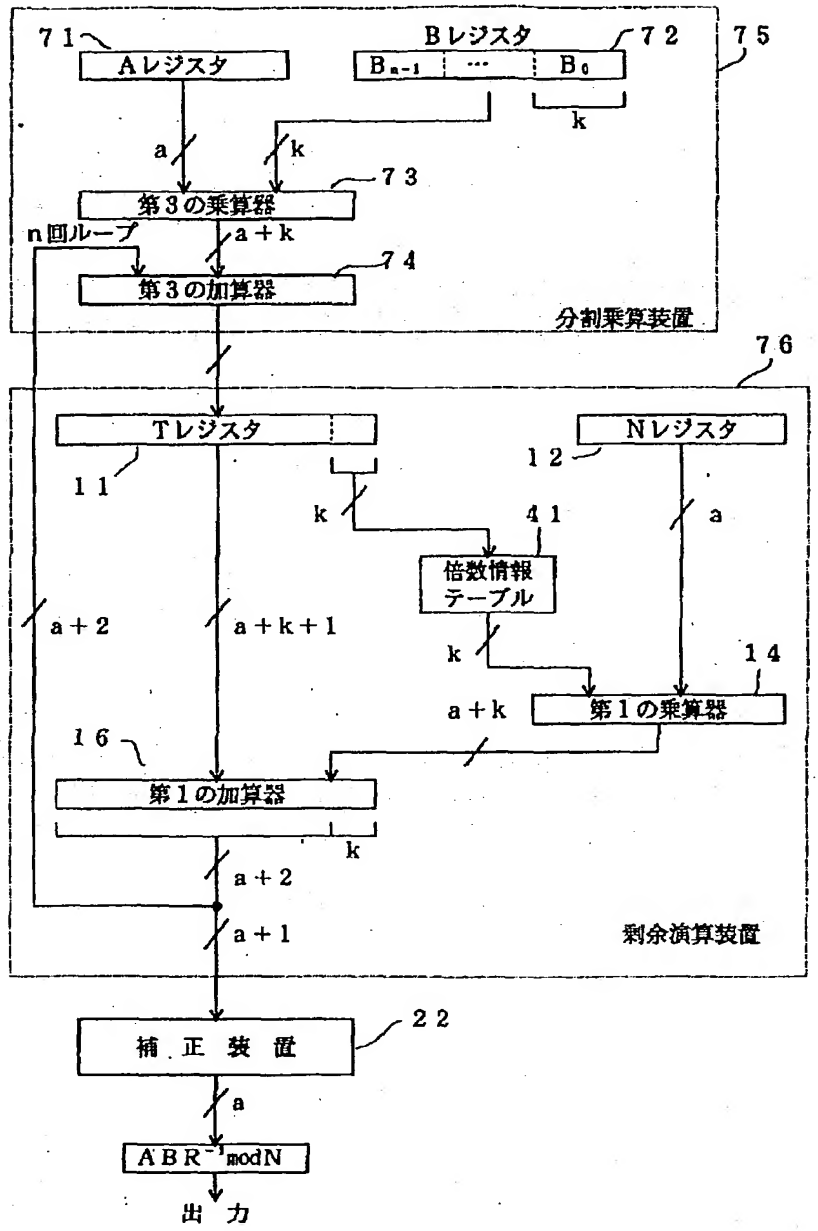


【図４】


倍数テーブル方式剰余計算装置の構成図
入　力
n回ループ
11
Tレジスタ
52
倍数算定部
k
51
上位
a+(n-1)k+1
a+nk
倍数テーブル
16
第１の加算器
k
a+nk+1
22
補正装置
第１のレジスタ
17
－Nレジスタ
18
a+1
a+1
a+1
19
第２の加算器
a+1
下位a
20
第２のレジスタ
a
21
1
0
1
セレクタ
a
TR⁻¹ modN
出　力
T→TR⁻¹ mod N　　R＝bⁿ　　b＝2ᵏ

【図６】

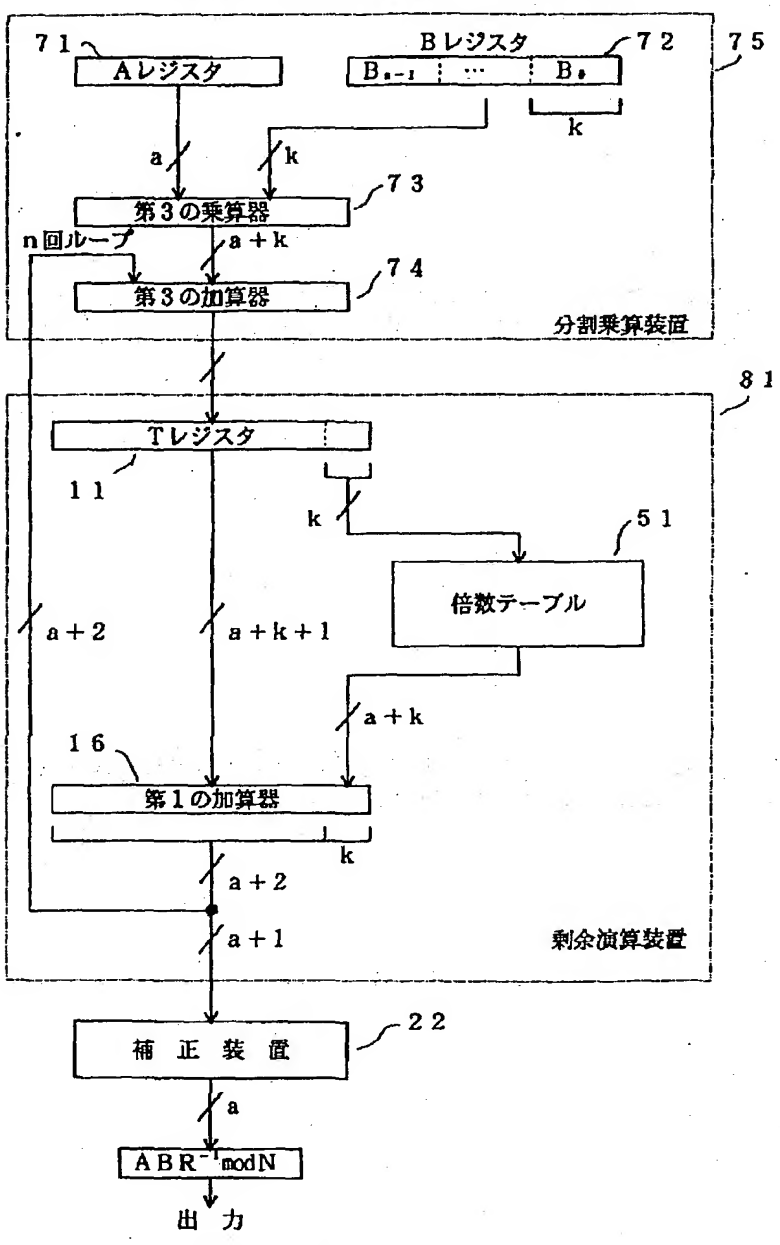
倍数情報テーブル方式乗算剰余計算装置の構成図



$AB \rightarrow ABR^{-1} \bmod N \qquad R = b^{a} \qquad b = 2^{k}$

【図７】

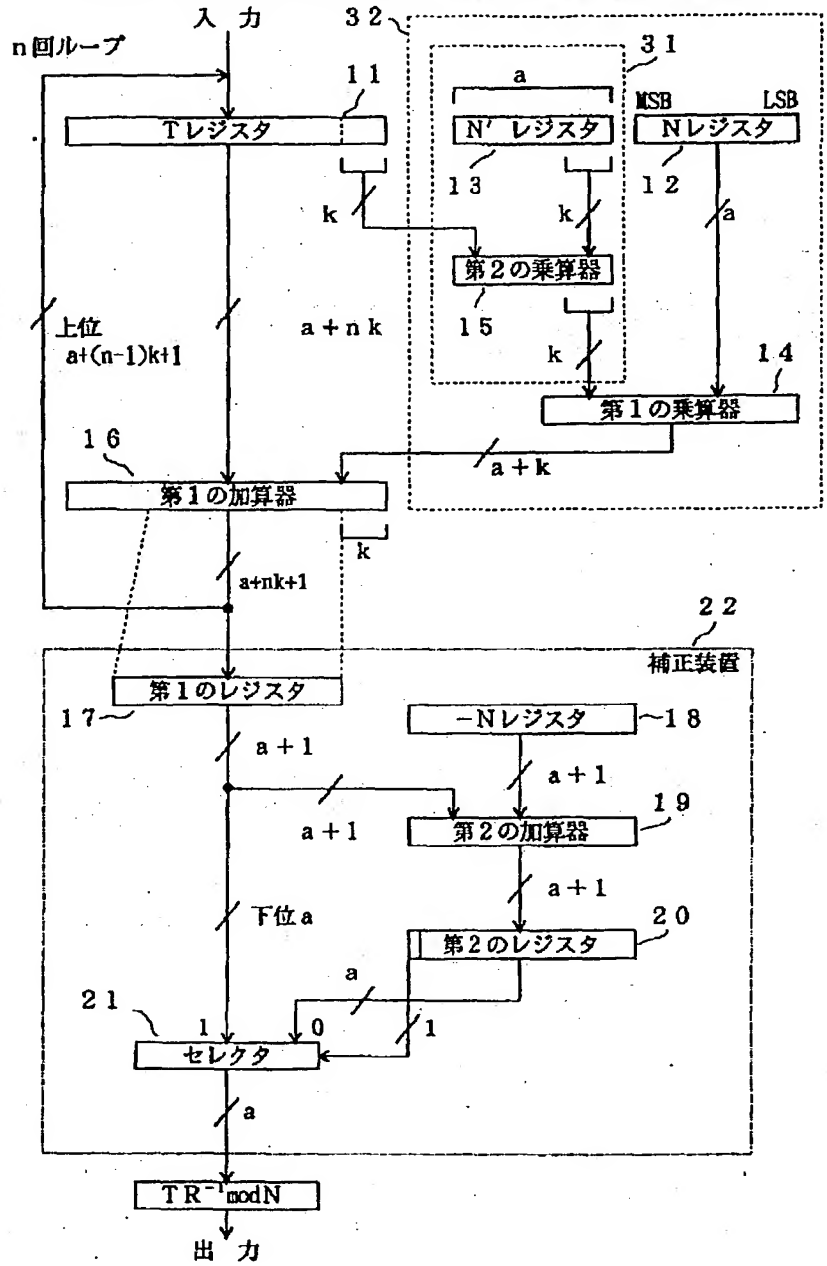
倍数テーブル方式乗算剰余計算装置の構成図



$AB \rightarrow ABR^{-1} \bmod N \qquad R = b^{a} \qquad b = 2^{k}$

【図８】

モンゴメリのアルゴリズムを用いた一般的な剰余計算装置の構成図



$T \rightarrow TR^{-1} \bmod N \qquad R = b^{n} \qquad b = 2^{k}$

フロントページの続き

(72)発明者　秋山　良太
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内